

CLAIMS

WHAT IS CLAIMED:

1. An apparatus for use in managing objects, the apparatus comprising:
a plurality of containers;
a plurality of clusters populating each container, each cluster comprising a plurality of objects;
a first data structure indicating a state of the objects; and
a second data structure indicating the state of the clusters.
2. The apparatus of claim 1, further comprising a plurality of container control data.
3. The apparatus of claim 1, wherein the container comprises a file.
4. The apparatus of claim 3, wherein the file is a page file or a swap file.
5. The apparatus of claim 3, wherein the objects comprise slots in the file.
6. The apparatus of claim 1, wherein each cluster comprises 16 objects.
7. The apparatus of claim 1, wherein at least one of the first and second data structures comprises a bitmap.
8. The apparatus of claim 1, further comprising at least one usage counter selected from the group consisting of:
a counter of how many free a cluster has;
a counter of how many free clusters are in the container;

5 a counter of how many sets of 2 adjacent bits are set in the words of the second data
6 structure;

7 a counter of how many sets of 4 adjacent bits are set in the words of the second data
8 structure;

9 a counter of how many sets of 8 adjacent bits are set in the words of the second data
10 structure;

11 a counter of how many sets of 16 adjacent bits are set in the words of the second data
12 structure;

13 a counter of how many sets of 32 adjacent bits are set in the words of the second data
14 structure; and

15 a counter of how many sets of 64 adjacent bits are set in the words of the second data
16 structure.

1 9. The apparatus of claim 8, further comprising at least one data structure
2 containing information indicating that:

3 the second data structure contains clusters of at least four adjacent free bits;

4 the second data structure is not empty, but contains no clusters of four adjacent free
5 bits;

6 the second data structure is empty, but allocation bitmap still shows free pages; or

7 the file is full or should not be used.

1 10. An apparatus, comprising:

2 a plurality of files;

3 a plurality of clusters populating each file, each cluster comprising a plurality of slots;

4 an allocation bitmap indicating a state of the slots; and

5 a directory bitmap indicating the state of the clusters.

1 11. The apparatus of claim 10, wherein each file is a page file or a swap file.

1 12. The apparatus of claim 10, wherein each cluster comprises 16 slots.

1 13. The apparatus of claim 10, further comprising at least one usage counter
2 selected from the group consisting of:

3 a counter of how many free slots a cluster has;

4 a counter of how many free clusters are in the file;

5 a counter of how many sets of 2 adjacent bits are set in the words of the directory
6 bitmap;

7 a counter of how many sets of 4 adjacent bits are set in the words of the directory
8 bitmap;

9 a counter of how many sets of 8 adjacent bits are set in the words of the directory
10 bitmap;

11 a counter of how many sets of 16 adjacent bits are set in the words of the directory
12 bitmap;

13 a counter of how many sets of 32 adjacent bits are set in the words of the directory
14 bitmap; and

15 a counter of how many sets of 64 adjacent bits are set in the words of the directory
16 bitmap.

1 14. The apparatus of claim 13, further comprising at least one data structure
2 containing information indicating that:

3 the directory bitmap contains clusters of at least four adjacent free bits;

4 the directory bitmap is not empty, but contains no clusters of four adjacent free bits;

5 the directory bitmap is empty, but the allocation bitmap still shows free pages; or

the file is full or should not be used.

15. A method for managing a plurality of clustered objects in a container, the method comprising:

tracking a state for each of a plurality of objects populating a container in a first data structure;

tracking a state for cluster of the memory like objects in a second data structure; and consulting at least one of the first and second data structures to manage the objects.

16. The method of claim 15, further comprising:

constructing the first data structure; and

constructing the second data structure.

17. The method of claim 15, wherein tracking the state for each of the plurality of objects populating the container in the first data structure or tracking the state for cluster of the memory like objects in the second data structure includes tracking in a bitmap.

18. The method of claim 15, further comprising consulting at least one usage counter to manage the objects.

19. The method of claim 18, wherein the consulting the at least one usage counter includes consulting a usage counter selected from the group consisting of:

a counter of how many free pages a file has;

a counter of how many free clusters are in the file;

a counter of how many sets of 2 adjacent bits are set in the words of the directory bitmap;

a counter of how many sets of 4 adjacent bits are set in the words of the directory bitmap;

9 a counter of how many sets of 8 adjacent bits are set in the words of the directory

10 bitmap;

11 a counter of how many sets of 16 adjacent bits are set in the words of the directory

12 bitmap;

13 a counter of how many sets of 32 adjacent bits are set in the words of the directory

14 bitmap; and

15 a counter of how many sets of 64 adjacent bits are set in the words of the directory

16 bitmap.

1 20. The method of claim 15, further comprising consulting at least one list
2 containing information extracted from usage counters to manage the objects.

1 21. The method of claim 20, wherein consulting the at least one list includes
2 consulting a list selected from the group consisting of:

3 a first list containing information indicating whether the second data structure
4 contains clusters of at least four adjacent free bits;

5 a second list containing information indicating whether the second data structure is
6 not empty, but contains no clusters of four adjacent free bits;

7 a third list containing information indicating whether the second data structure is
8 empty, but allocation bitmap still shows free objects; or

9 a fourth list containing information indicating that the containers in this list are full or
10 should not be used.

1 22. A program storage medium encoded with instructions that, when executed by
2 a computing device, perform a method for managing a plurality of clustered objects in a
3 container, the method comprising:

4 tracking a state for each of a plurality of objects populating a container in a first data
5 structure;
6 tracking a state for cluster of the objects in a second data structure; and
7 consulting at least one of the first and second data structures to manage the objects.

1 23. The program storage medium of claim 22, wherein the encoded method
2 further comprises:

3 constructing the first data structure; and
4 constructing the second data structure.

1 24. The program storage medium of claim 22, wherein tracking the state for each
2 of the plurality of objects populating the container in the first data structure or tracking the
3 state for cluster of the memory like objects in the second data structure in the encoded
4 method includes tracking in a bitmap.

1 25. The program storage medium of claim 22, wherein the encoded method
2 further comprises consulting at least one usage counter to manage the objects.

1 26. The program storage medium of claim 25, wherein the consulting the at least
2 one usage counter in the encoded method includes consulting a usage counter selected from
3 the group consisting of:

4 a counter of how many free pages a file has;
5 a counter of how many free clusters are in the file;
6 a counter of how many sets of 2 adjacent bits are set in the words of the directory
7 bitmap;
8 a counter of how many sets of 4 adjacent bits are set in the words of the directory
9 bitmap;

10 a counter of how many sets of 8 adjacent bits are set in the words of the directory
11 bitmap;
12 a counter of how many sets of 16 adjacent bits are set in the words of the directory
13 bitmap;
14 a counter of how many sets of 32 adjacent bits are set in the words of the directory
15 bitmap; and
16 a counter of how many sets of 64 adjacent bits are set in the words of the directory
17 bitmap.

1 27. The program storage medium of claim 22, wherein the encoded method
2 further comprises consulting at least one list containing information extracted from usage
3 counters to manage the objects.

1 28. The program storage medium of claim 27, wherein consulting the at least one
2 list in the encoded method includes consulting a list selected from the group consisting of:

3 a first list containing information indicating whether the second data structure
4 contains clusters of at least four adjacent free bits;
5 a second list containing information indicating whether the second data structure is
6 not empty, but contains no clusters of four adjacent free bits;
7 a third list containing information indicating whether the second data structure is
8 empty, but the allocation bitmap still shows free objects; or
9 a fourth list containing information indicating that the containers in this list are full or
10 should not be used.

1 29. A computing device programmed to perform a method for managing a
2 plurality of clustered objects in a container, the method comprising:

tracking a state for each of a plurality of objects populating a container in a first data structure;

tracking a state for cluster of the memory like objects in a second data structure; and consulting at least one of the first and second data structures to manage the objects.

30. The programmed computing device of claim 29, wherein the programmed method further comprises:

constructing the first data structure; and
constructing the second data structure.

31. The programmed computing device of claim 29, wherein tracking the state for each of the plurality of objects populating the container in the first data structure or tracking the state for cluster of the memory like objects in the second data structure in the programmed method includes tracking in a bitmap.

32. The programmed computing device of claim 29, wherein the programmed method further comprises consulting at least one usage counter to manage the objects.

33. The programmed computing device of claim 32, wherein the consulting the at least one usage counter in the programmed method includes consulting a usage counter selected from the group consisting of:

a counter of how many free pages a file has;
a counter of how many free clusters are in the file;
a counter of how many sets of 2 adjacent bits are set in the words of the directory
bitmap;
a counter of how many sets of 4 adjacent bits are set in the words of the directory
bitmap;

a counter of how many sets of 8 adjacent bits are set in the words of the directory

bitmap;

a counter of how many sets of 16 adjacent bits are set in the words of the directory

bitmap;

a counter of how many sets of 32 adjacent bits are set in the words of the directory

bitmap; and

a counter of how many sets of 64 adjacent bits are set in the words of the directory

bitmap.

34. The programmed computing device of claim 29, wherein the programmed method further comprises consulting at least one list containing information extracted from usage counters to manage the objects.

35. The programmed computing device of claim 34, wherein consulting the at least one list in the programmed method includes consulting a list selected from the group consisting of:

a first list containing information indicating whether the second data structure contains clusters of at least four adjacent free bits;

a second list containing information indicating whether the second data structure is not empty, but contains no clusters of four adjacent free bits;

a third list containing information indicating whether the second data structure is empty, but allocation bitmap still shows free objects; or

a fourth list containing information indicating that the containers in this list are full or should not be used.

36. A method for managing a plurality of clustered slots in a file, the method comprising:

3 tracking a state for each of a plurality of slots populating a file in a allocation data
4 structure;

5 tracking a state for cluster of the memory like objects in a directory data structure; and
6 consulting at least one of the first and directory data structures to manage the slots.

1 37. The method of claim 36, further comprising:
2 constructing the allocation data structure; and
3 constructing the directory data structure.

1 38. The method of claim 36, wherein tracking the state for each of the plurality of
2 slots populating the file in the allocation data structure or tracking the state for cluster of the
3 memory like objects in the directory data structure includes tracking in a bitmap.

1 39. The method of claim 36, further comprising consulting at least one usage
2 counter to manage the slots.

1 40. The method of claim 39, wherein the consulting the at least one usage counter
2 includes consulting a usage counter selected from the group consisting of:

3 a counter of how many free pages a file has;

4 a counter of how many free clusters are in the file;

5 a counter of how many sets of 2 adjacent bits are set in the words of the directory
6 bitmap;

7 a counter of how many sets of 4 adjacent bits are set in the words of the directory
8 bitmap;

9 a counter of how many sets of 8 adjacent bits are set in the words of the directory
10 bitmap;

11 a counter of how many sets of 16 adjacent bits are set in the words of the directory
12 bitmap;
13 a counter of how many sets of 32 adjacent bits are set in the words of the directory
14 bitmap; and
15 a counter of how many sets of 64 adjacent bits are set in the words of the directory
16 bitmap.

1 41. The method of claim 36, further comprising consulting at least one list
2 containing information extracted from usage counters to manage the slots.

1 42. The method of claim 41, wherein consulting the at least one list includes
2 consulting a list selected from the group consisting of:

3 a first list containing information indicating that the directory data structure for files
4 in this list contains clusters of at least four adjacent free bits;
5 a second list containing information indicating that the directory data structure for
6 files in this list is not empty, but contains no clusters of four adjacent free bits;
7 a third list containing information indicating that the directory data structure for files
8 in this list is empty, but allocation bitmap still shows free slots; or
9 a fourth list containing information indicating that files in this list are full or should
10 not be used.

1 43. A program storage medium encoded with instructions that, when executed by
2 a computing device, perform a method for managing a plurality of clustered slots in a file, the
3 method comprising:

4 tracking a state for each of a plurality of slots populating a file in a allocation data
5 structure;
6 tracking a state for cluster of the memory like objects in a directory data structure; and

consulting at least one of the first and directory data structures to manage the slots.

44. The program storage medium of claim 43, wherein the encoded method further comprises:

constructing the allocation data structure; and

constructing the directory data structure.

45. The program storage medium of claim 43, wherein tracking the state for each of the plurality of slots populating the file in the allocation data structure or tracking the state for cluster of the memory like objects in the directory data structure in the encoded method includes tracking in a bitmap.

46. The program storage medium of claim 43, wherein the encoded method further comprises consulting at least one usage counter to manage the slots.

47. The program storage medium of claim 46, wherein the consulting the at least one usage counter in the encoded method includes consulting a usage counter selected from the group consisting of:

a counter of how many free slots a file has;

a counter of how many free clusters are in the file;

a counter of how many sets of 2 adjacent bits are set in the words of the directory
bitmap;

a counter of how many sets of 4 adjacent bits are set in the words of the directory
bitmap;

a counter of how many sets of 8 adjacent bits are set in the words of the directory
bitmap;

12 a counter of how many sets of 16 adjacent bits are set in the words of the directory
13 bitmap;
14 a counter of how many sets of 32 adjacent bits are set in the words of the directory
15 bitmap; and
16 a counter of how many sets of 64 adjacent bits are set in the words of the directory
17 bitmap.

1 48. The program storage medium of claim 43, wherein the encoded method
2 further comprises consulting at least one list containing information extracted from usage
3 counters to manage the slots.

1 49. The program storage medium of claim 48, wherein consulting the at least one
2 list in the encoded method includes consulting a list selected from the group consisting of:

3 a first list containing information indicating whether the directory data structure
4 contains clusters of at least four adjacent free bits;
5 a second list containing information indicating whether the directory data structure is
6 not empty, but contains no clusters of four adjacent free bits;
7 a third list containing information indicating whether the directory data structure is
8 empty, but allocation bitmap still shows free slots; or
9 a fourth list containing information indicating whether the file is full or should not be
10 used.

1 50. A computing device programmed to perform a method for managing a
2 plurality of clustered slots in a file, the method comprising:

3 tracking a state for each of a plurality of slots populating a file in a allocation data
4 structure;
5 tracking a state for cluster of the memory like objects in a directory data structure; and

6 consulting at least one of the first and directory data structures to manage the slots.

1 51. The programmed computing device of claim 50, wherein the programmed
2 method further comprises:

3 constructing the allocation data structure; and

4 constructing the directory data structure.

1 52. The programmed computing device of claim 50, wherein tracking the state for
2 each of the plurality of slots populating the file in the allocation data structure or tracking the
3 state for cluster of the memory like objects in the directory data structure in the programmed
4 method includes tracking in a bitmap.

1 53. The programmed computing device of claim 50, wherein the programmed
2 method further comprises consulting at least one usage counter to manage the slots.

1 54. The programmed computing device of claim 53, wherein the consulting the at
2 least one usage counter in the programmed method includes consulting a usage counter
3 selected from the group consisting of:

4 a counter of how many free slots a file has;

5 a counter of how many free clusters are in the file;

6 a counter of how many sets of 2 adjacent bits are set in the words of the directory
7 bitmap;

8 a counter of how many sets of 4 adjacent bits are set in the words of the directory
9 bitmap;

10 a counter of how many sets of 8 adjacent bits are set in the words of the directory
11 bitmap;

12 a counter of how many sets of 16 adjacent bits are set in the words of the directory
13 bitmap;
14 a counter of how many sets of 32 adjacent bits are set in the words of the directory
15 bitmap; and
16 a counter of how many sets of 64 adjacent bits are set in the words of the directory
17 bitmap.

55. The programmed computing device of claim 50, wherein the programmed
method further comprises consulting at least one list containing information extracted from
usage counters to manage the slots.

56. The programmed computing device of claim 55, wherein consulting the at
least one list in the programmed method includes consulting a list selected from the group
consisting of:

- a first list containing information indicating whether the directory data structure
contains clusters of at least four adjacent free bits;
- a second list containing information indicating whether the directory data structure is
not empty, but contains no clusters of four adjacent free bits;
- a third list containing information indicating whether the directory data structure is
empty, but allocation bitmap still shows free slots; or
- a fourth list containing information indicating whether the file is full or should not be
used.